JOHANNES KEPLER
UNIVERSITY LINZ

Bachelor's Thesis

Automatic Detection of Data Structures

in Reconstructed Heap States

Student: Manuel Vujakovic

Advisor: Dipl.-Ing. Dr. Markus Weninger, BSc

Start date: February 2022

Dipl.-Ing. Dr. Markus Weninger, BSc
Institute for System Software
P +43-732-2468-4361
markus.weninger@jku.at

AntTracks, developed at the Institute for System Software at the Johannes Kepler University, is a memory monitoring tool that is used to detect and analyze memory anomalies such as memory leaks. Often, memory leaks are related to data structure misuse, e.g., programming errors leading to growing data structures (such as lists, maps, sets, etc.). Yet, memory analysis tools are by default not aware of the concept of data structures but only see the heap as an object graph, i.e., separate objects that reference each other (thereby keeping each other alive). Since modern applications may contain more than 100 million objects in their heap, analysis on the object level is often too fine-grained and complex.

The goal of this project is to automatically identify data structures in the heap, for example, to automatically identify that `LinkedList` objects are data structures that internally consist of `LinkedList$Node` objects that point to data objects. Such detected data structures can then be used to perform less fine-grained analyses, instead providing a more top-level view of the memory behavior of the application. It is not a goal of this thesis to develop new analysis approaches bases on detected data structures.

To become familiar with automatic data structure detection, the student should thoroughly explore and summarize existing literature on data structure detection and aggregation. Existing approaches often classify heap objects based on their roles in data structures, such as “recursive backbones” or “array backbones” (for example “*Patterns of Memory Inefficiency*” by *Chis et al., https://doi.org/10.1007/978-3-642-22655-7_18* and various work by *Mitchell et al.*). Yet, such approaches often rely on external information which tells them, for example, where data structures start, i.e., which objects are “data structure heads” (for example “*Analyzing Growth Over Time to Facilitate Memory Leak Detection”* by *Weninger et al., https://doi.org/10.1145/3297663.3310297)*.

The thesis should contain a comprehensive presentation of existing approaches, discussing their similarities and differences. Based on these observations, the student should devise at least one own novel detection algorithm that focuses on not relying on external information. The algorithm should use suitable heuristics to decide where data structures start, which other objects are “internal” parts of the data structure, and where data structures end. Ideas for such an algorithm should regularly be discussed with the supervisor. Finally, the algorithm should be implemented as a prototype in AntTracks to show its feasibility and applicability.
The thesis should present how the new algorithm works in general, and which kinds of data structures can be detected with it. More specifically, corner cases should be discussed. For example, the thesis should highlight how the devised algorithm behaves when deciding which object is a data structure head (for example, in Java, a `HashSet` internally contains a `HashMap` to store its data, and existing approach often detect the `HashMap` as data structure, but not the overall `HashSet`). Also, the thesis should present how the algorithm handles data structures that are contained in other data structures (such as a HashMap that in turn contains `ArrayLists` as values).

Modalities:
The progress of the project should be discussed at least every three weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks and discussed with the advisor and the supervisor. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than 30.09.2022.